PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 :
G06F 15/403, 15/40

A1

(11) International Publication Number: WO 91/03024

(43) International Publication Date: 7 March 1991 (07.03.91)

(21) International Application Number: PCT/US90/04570

(22) International Filing Date: 14 August 1990 (14.08.90)

(30) Priority data:
394,324 14 August 1989 (14.08.89) US

(71) Applicant: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, WA 98052-6399 (US).

(72) Inventors: MASDEN, Kenneth, E. ; 14504 N.E. 77th Street, Redmond, WA 98052 (US). SMIRL, John, D. ; 8215 N.E. 115th Place, Kirkland, WA 98034 (US). KING, John, W. ; 1942 3rd Street, Kirkland, WA 98033 (US). RUBIN, Darryl, E. ; 17707 N.E. 141st St., Redmond, WA 98052 (US).

(74) Agents: JACOBSON, James, E., Jr. et al.; Seed and Berry, 6300 Columbia Center, Seattle, WA 98104-7092 (US).

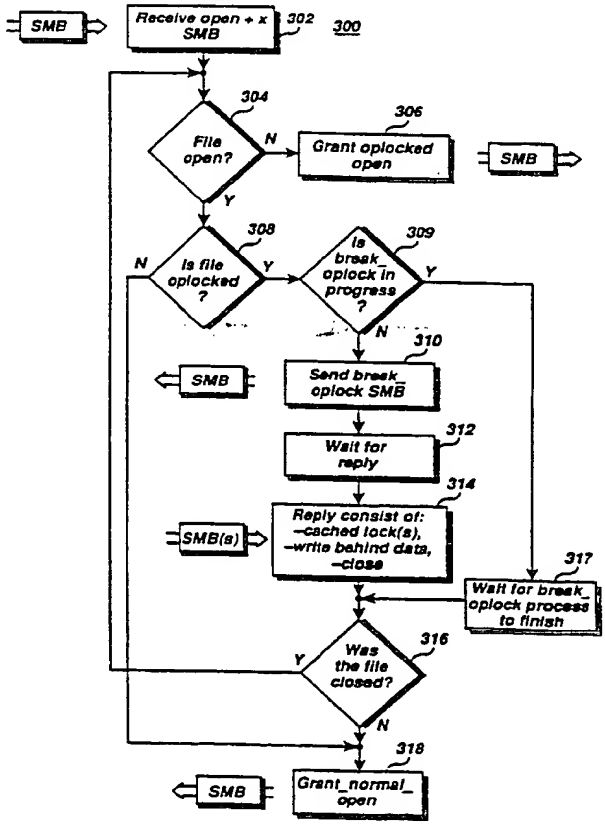
(81) Designated States: AT (European patent), AU, BB, BE (European patent), BF (OAPI patent), BG, BJ (OAPI patent), BR, CA, CF (OAPI patent), CG (OAPI patent), CH (European patent), CM (OAPI patent), DE (European patent)*, DK (European patent), ES (European patent), FI, FR (European patent), GA (OAPI patent), GB (European patent), HU, IT (European patent), JP, KR, LK, LU (European patent), MC, MG, ML (OAPI patent), MR (OAPI patent), MW, NL (European patent), NO, RO, SD, SE (European patent), SN (OAPI patent), SU, TD (OAPI patent), TG (OAPI patent).

Published
*With international search report.*
*Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.*

(54) Title: METHOD AND SYSTEM FOR OPPORTUNISTIC LOCKING IN A NETWORKED COMPUTER SYSTEM

(57) Abstract

The present invention contemplates a method and means for improving the performance of distributed computer systems including a file server and a plurality of remote workstations. In operation, a request is issued from one of the remote workstations to open a file or record resident on the file server in a sharing mode. The system of the present invention automatically converts the sharing mode request into a request to open the file or record in an opportunistic locked exclusive mode. The file server then determines whether the requested file or record has been opened by another workstation. If the file or record is not currently open, the file or record is opened in the opportunistic locked exclusive mode and the file or record is processed and cached locally in a system workstation. If the file or record is already open, the file is opened in the conventional sharing mode. If the file or record opened in the opportunistic locked exclusive mode is requested by another client, the locally cached data is written to the file server and the system reverts to the conventional sharing mode. The system of the present invention is also compatible with conventional exclusive mode operation which may be manually requested by a client.




* See back of page

## DESIGNATIONS OF "DE"

Until further notice, any designation of "DE" in any international application whose international filing date is prior to October 3, 1990, shall have effect in the territory of the Federal Republic of Germany with the exception of the territory of the former German Democratic Republic.

*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | |
|---|---|
| AT | Austria |
| AU | Australia |
| BB | Barbados |
| BE | Belgium |
| BF | Burkina Faso |
| BG | Bulgaria |
| BJ | Benin |
| BR | Brazil |
| CA | Canada |
| CF | Central African Republic |
| CG | Congo |
| CH | Switzerland |
| CM | Cameroon |
| DE | Germany |
| DK | Denmark |
| ES | Spain |
| FI | Finland |
| FR | France |
| GA | Gabon |
| GB | United Kingdom |
| GR | Greece |
| HU | Hungary |
| IT | Italy |
| JP | Japan |
| KP | Democratic People's Republic of Korea |
| KR | Republic of Korea |
| LI | Liechtenstein |
| LK | Sri Lanka |
| LU | Luxembourg |
| MC | Monaco |
| MG | Madagascar |
| ML | Mali |
| MR | Mauritania |
| MW | Malawi |
| NL | Netherlands |
| NO | Norway |
| PL | Poland |
| RO | Romania |
| SD | Sudan |
| SE | Sweden |
| SN | Senegal |
| SU | Soviet Union |
| TD | Chad |
| TG | Togo |
| US | United States of America |

BNSDOCID: <WO___9103024A1_I_>

Description

# METHOD AND SYSTEM FOR OPPORTUNISTIC LOCKING IN A NETWORKED COMPUTER SYSTEM

## Field of the Invention

This invention relates to the field of computer networks and, more specifically, to a method and means for improving the speed and performance of a networked computer system.

## Background of the Invention

Computer networks are a well-known computer architecture wherein users of the network are typically located in many diverse locations. Computer networks are adapted for use with independent computer systems, each having local file storage capability, as well as diskless terminal-type systems wherein master files are stored on a single file server.

One popular use for distributed computer systems is database management. In a multiuser environment, it is important that all users of the database have access to current information as each user of the system is permitted to make changes to the database at will. Therefore, database files are typically resident on a single-system file server. When accessing a database with remote workstations, the remote workstations typically issue requests to the system file server to retrieve the desired information. As an entry in the file is updated or modified, the new information is sent from the workstation to the file server so that all "clients" in the system have access to current information. For example, in mulituser database applications, many clients have access to all records in the database. If a client reads and updates a record, and a second client reads the same record before the master file is updated, each client will have

inconsistent versions of the same record. Most database applications include some provision to avoid the problem of inconsistent data. Typically, while operating in a sharing mode, a client issues a request to open and lock specified records in the database. The specified records are then read and modified. The modified records are then written to the master file on the server and the records are then unlocked and closed. While the records are locked, no other client has access to the record. While this technique prevents inconsistent data, the process of opening, locking, reading, writing, closing, and unlocking files or records may result in as many as eight messages across the computer network.

In many computer networks, it is common for only a single user to be using the network at any one time and inconsistent data is not possible. In this situation, the process of opening, locking, reading, writing, unlocking, and closing files or records results in wasted time and increased system overhead.

The present invention overcomes the above-mentioned problem by allowing network workstations to automatically operate in an opportunistic locked exclusive mode when files are only used by single clients and by automatically switching to a conventional sharing mode when file data is required by multiple clients. In the opportunistic locked exclusive mode, files or records are processed and cached locally, thus eliminating a large portion of network messaging traffic. If another client requires access to the file, the locally cached data is written to the file server, and the system reverts to sharing mode operation.

## Summary and Objects of the Invention

In summary, the present invention contemplates a method and means for improving the performance of distributed computer systems including a file server and

BNSDOCID: <WO___9103024A1_I_>

a plurality of remote workstations. In operation, a request is issued from one of the remote workstations to open a file or record resident on the file server in a sharing mode. The system of the present invention automatically converts the sharing mode request into a request to open the file or record in an opportunistic locked exclusive mode. The file server then determines whether the requested file or record has been opened by another workstation. If the file or record is not currently open, the file or record is opened in the opportunistic locked exclusive mode and the file or record is processed and cached locally in a system workstation. If the file or record is already open, the file is opened in the conventional sharing mode. If the file or record opened in the opportunistic locked exclusive mode is requested by another client, the locally cached data is written to the file server and the system reverts to the conventional sharing mode. The system of the present invention is also compatible with conventional exclusive mode operation which may be manually requested by a client.

Accordingly, it is an object of the present invention to provide a method and means for improving the performance of a networked computer system.

It is another object of the present invention to eliminate the system overhead of workstations operating in a sharing mode when file services are only required by a single workstation at a given point in time.

It is another object of the present invention to provide a method and means for automatically switching workstations between an opportunistic locked exclusive mode and a sharing mode based on the number of users requiring access to files resident on a file server.

It is another object of the present invention to provide opportunistic exclusive locked mode operation

BNSDOCID <WO _ 9103024A1 I >

which is transparently available to application programs without modification of the application program.

Brief Description of the Drawings

These and other objects may be fully understood through the detailed description below and the accompanying drawings in which:

Figure 1 is a diagram of a typical local area network system.

Figure 2 is a more detailed diagram of the system of Figure 1.

Figure 3 is a flow diagram detailing the operation of the file server of the present invention when opening file server files.

Figure 4 is a flow diagram which describes the operation of the workstations of the present invention when requesting a file to be opened in the OPlocked exclusive mode.

Figure 5 is a flow diagram which describes the operation of the workstations of the present invention when the OPlocked exclusive mode operation must be terminated.

Detailed Description of the Invention

While the principles of the present invention may be applied to virtually any distributed computer system, the present invention is particularly adapted for use with local area network (LAN) systems of the type shown in Figure 1. LAN systems are particularly adapted for multiuser applications, one example of which are database management systems. In the system 100, a file server 102 provides file services to a plurality of remote workstations 104, 106, 108 and 110. While the system 100 is shown with four workstations, those skilled in the art will appreciate that LAN systems are adapted for use with virtually any number of workstations. Furthermore, the system 100 is shown with

each workstation directly coupled to the file server 102. The present invention is adapted for use with all types of networked systems regardless of the method used to interface the file server and workstations. In the context of the present invention, users of the LAN system are referred to as clients and file services refers to various file management tasks such as opening, updating and closing files. In the context of the present invention, the terms "file" or "files" refers to any collection of information or data regardless of size or whether the information or data is merely a portion or subset of a larger collection of information or data. The principles of the present invention may be applied to any collection of information accessed in a distributed manner.

A more detailed diagram of the system 100 is shown in Figure 2. While the present invention is described herein in the context of a LAN system, the principles of the present invention may be applied to any type of distributed computer system. The present invention is adapted to operate with many known LAN systems and the LAN system 100 is preferably modelled after the well known LAN architecture described in ISO specification ISO 7498, entitled "Open Systems Interconnection - Basic Reference Model." LAN system principles are also described in a text: Tannebaum, "Computer Networks," Prentice-Hall, Inc., 1981, ISBN-0-13-165183-8. The LAN system principles described in the above-mentioned publications are well known to persons of ordinary skill in the art. Briefly, however, The file server 102 comprises a server process 202 which cooperates with a kernel 204 to receive, generate and process system requests. The kernel further cooperates with a mechanism 206 which provides storage for system files. The kernel 204 communicates with the network through a redirector 208 and a transport 210. The redirector 208 and transport 210 are responsible for

.BNSDOCID: <WO__9103024A1_I_>

monitoring message traffic on the LAN, for communicating LAN messages to the kernel 204 and server process 202, and for packaging and returning messages or data from the kernel 204 and server process 202 to the LAN.

The file server 102 is coupled to LAN communication lines 212 through a LAN interface 214. The LAN interface converts the messages and data processed by the file server 102 into a form consistent with the communications protocol of the LAN network. One such LAN protocol is Ethernet. A plurality of workstations 104, 106, 110 are further coupled to LAN communication lines through LAN interface devices 216, 218, and 220, respectively. Each respective workstation comprises a transport 222, a redirector 224, a kernel 226 and application programs 228. It is anticipated that each workstation used with the LAN system is capable of independent or local operations as well as being able to access and modify files resident on file server 102.

In prior LAN systems, workstations are operated in a sharing mode wherein all files are maintained on a file server, and are accessible to all clients simultaneously, or in an exclusive mode wherein files, or portions thereof, are transferred from the file server to the workstation for local processing. In the sharing mode, workstations issue requests to the file server to open and lock files or records resident on the file server. The file server responds by verifying the transaction and sending the requested data back to the workstation. The data is then modified by the workstation and then immediately written back to the file server. This is followed by an instruction from the workstation to unlock and close the file. Therefore, in the sharing mode, every time a record is modified by a workstation, as many as eight messages may be communicated between the workstation and the file server. This model is based on the assumption that many

workstations may be working with a single file at the same time and thus, all portions of the file must be maintained at a single location, i.e., the file is typically resident on the file server so that all clients in the system have access to the most recent modifications to file records. While individual records are being modified by clients, they are temporarily locked to prevent inconsistent data.

In the exclusive mode of prior systems, clients may manually instruct the file server to open a file exclusively, and deny access to all other users. The file (or portions of the file) is then transferred to the workstation requesting the file where it remains until the client closes the file and returns it to the file server. While this eliminates the need for communicating with the file server while operating in the exclusive mode, no other client in the system has access to the file while in this mode. Therefore, file access is provided on a first come, first served basis and many important file service requests may be denied. While the present invention is disclosed as an improvement to operating systems, the principles of the present invention are equally applicable to cooperating application systems. Furthermore, since the present invention may be incorporated in an operating system, existing applications may employ the present invention without modification of the application.

In most LAN networks, the probability that a file is being modified only by a single user is quite high, but multiuser application problems must be written to assume multiple users. Therefore, the communication between a single workstation and the file server when operating in the sharing mode results in wasted time and increased system overhead. However, if a client opens a file in the exclusive mode, important file service requests from other clients may be denied. The present invention overcomes this problem by providing

BNSDOCID: <WO__9103024A1_I_>

workstations which are adapted to operate in three modes: an opportunistic locked (OPlocked) exclusive mode, a sharing mode and a conventional exclusive mode. For the purpose of simplifying the explanation of the operation of the present invention, the explanation herein refers to the transfer of files. Those skilled in the art will appreciate that complete files are rarely transferred and in most applications, only portions of files are actually transferred between the file server and the associated workstations.

As in prior systems, in the sharing mode it is assumed that a plurality of users may be modifying records in a single file simultaneously and the master file is not moved from the file server. In the OPlocked exclusive mode of the present invention, a single file (or a portion thereof) may be transferred to a workstation to be processed and cached locally and the file is automatically OPlocked. When the user is completed with the file, the updated file is then returned to the file server. However, if another workstation requests file services on an Oplocked file, the file server sends a message to the workstation currently owning the file to return the updated file to the file server and to revert to a sharing mode.

The present invention eliminates the problem of denying access to locked files while providing the advantages of exclusive mode operation by allowing automatic exclusive mode operation as long as no other clients require file services for that file. In yet another aspect of the present invention, when a workstation requests sharing mode file services for a file, a sharing mode request is converted into a request for operation in the Oplocked exclusive mode. The file server then determines whether the file is currently open. If not, the file server opens the file in the Oplocked exclusive mode, even if exclusive mode operation is not requested by the client Application.

BNSDOCID: <WO___9103024A1_I_>

Therefore, file services are efficiently provided to clients in a manner which is totally transparent to the client. The present invention is totally compatible with the manual exclusive mode operation of prior systems. While the present invention is illustrated as a modification to workstation and file server operating systems, those skilled in the art will appreciate that the present invention could also be implemented within cooperating application programs.

The following pseudo-code describes the operation of the present invention.

Workstation:

```
If file open request is for sharing_mode then
     send server request for open with OPlock
else (open is for exclusive mode)
     send server a normal open request
     OPlock does not apply
```

When the server returns the open request the server indicates if the file has been Oplocked.

```
If the file was not Oplocked then
     proceed as normal for a sharing mode open
else (the file is Oplocked)
     Treat the file as an exclusive open.
```

The workstation can perform:

(a) read ahead, write behind, lock caching, etc.

(b) Any operation that could previously only be used by an exclusive open is used on an Oplocked exclusive open.

While the file is Oplocked, the server may dynamically disallow the Oplocked status if file

BNSDOCID: <WO__9103024A1_I_>

services are requested by another workstation. This is done by sending a break_oplock message from the server to the workstation that has the file Oplocked.

When a break_oplock message is received:

If the workstation has the file open:

a) flush any write_behind data to the server
b) flush any cached locks to the server
c) invalidate any read ahead data
d) send an oplock_broken message to the server

The workstation still has the file open. However, the workstation can't use any of the buffering algorithms (read ahead, etc) until the workstation reverts to an Oplocked mode.

If the workstation receives a break_oplock message for a file that is not opened, it is assumed that the workstation's close of the file crossed the break_oplock message on the communication lines and the break_oplock message is ignored.

Server:

When the server receives an open with OPlock request:

1) if this is the first client opening the file, the OPlock request is granted.

2) if this is the second client requesting an open of the file, and the first client has the file Oplocked, the file server sends a break OPlock message to the workstation holding the Oplocked file. This workstation then sends any write_behind data and any cached locks to the server. This data is followed by an oplock_broken

message. After the oplock_broken message is received by the server, the server sends the second client an open but not Oplocked response.

The workstation holding the OPlock is not required to respond with a break OPlock message. Instead the workstation may simply respond by closing the file. In this case the second client will now be the only opener and will receive the OPlock.

3) If any other opens are received while a break OPlock is in progress, these open requests are queued until the break OPlock operation is complete. At this time they will be processed as normal though steps 1 and 2. The result will be sending opened but not Oplocked responses to any open that was pending.

where:

**read ahead data:** In accordance with the principles of the present invention, if a client requests, for example, ten bytes of data, a workstation local buffer is set up, i.e., 4K in size, and 4K of data is read into the local buffer. The buffer is set up in the workstation redirector with the expectation that the user will eventually need this data. The next time a request for the data is made, it can be handled locally without accessing the file server. This is used in the exclusive mode and the Oplocked exclusive mode.

**write behind data:** In accordance with the principles of the present invention, if the client writes or saves small amounts of data it is stored in a workstation local buffer until buffer is full. Once the buffer is filled, the data is transmitted to the server. This is used in the exclusive mode and the Oplocked exclusive mode.

**lock caching:** The system of the present invention tracks locks placed on a file locally. If it is known that a file is OPlocked, the request is not sent to the file server. If in the sharing mode, the locked bytes are sent to the file server. When in the sharing mode an application may make a call to lock certain bytes while they are being modified. For example, in the sharing mode, locked requests are sent to the file server to get certain records to modify. Once the records are modified, the records are send back to the file server with an unlock request. File lock and unlock requests are a feature of the well known MS-DOS operating system as well as many other operating systems.

**cached locks:** locks that are present locally in a workstation and haven't been sent to the file server. In other words, the client believes the locked range is locked.

The OPlocking process of the present invention is further described in conjunction with Figures 3 - 5. Figure 3 is a flow diagram of the operation of the file server portion of the system operating in accordance with the principles of the present invention. As will be discussed below, the present invention is adapted for use with the Server Message Block (SMB) protocol developed by Microsoft, Intel and IBM. For the purposes of the discussion below, SMBs refer to message packets sent between the server and workstations over the LAN communication lines. The process 300 is invoked when an open file SMB is received from a system workstation as indicated by item 302. The open file SMB includes a request to open the file in the OPlocked exclusive mode. Control then passes to decision 304 to determine whether the requested file has already been opened by another

workstation. If not, item 306 is invoked to generate an SMB granting the file open request in a exclusive OPlocked mode.

SMB requests typically include a desired operation such as open, as well as another operation such as a request to read a designated block of data. For example, an SMB request may be a request to open and read, open and write, read and lock, etc. The SMB message returned then returns the SMB instruction along with the file handle and the requested data. The present invention is directed primarily to the opening and closing of files and the mode in which files are opened. Therefore, the additional information contained in an SMB packet is not further discussed herein. Once a return SMB is generated in item 306, the SMB is sent to the requesting workstation via the LAN communication lines. If the requested file is currently open, control passes to decision 308 to determine whether the open file is OPlocked. If not, control is passed to item 318 to generate a "grant normal open" SMB. If decision 308 determines the file is OPlocked, control is passed to decision 309 to determine whether a break_oplock function is in process. If so, control is passed to item 317 to wait until the break_oplock process is complete. Once complete, control passes to decision 316, then to item 318 to generate a grant_normal_open SMB. If decision 309 determines a break_oplock function is not in process, control is passed to item 310 to generate and send a "break OPlock" SMB to the original file opener. Control then passes to item 312 to wait for a reply SMB. The reply SMB is received by item 314 wherein the reply may consist of any combination of cached locks, write behind data or a close instruction. Control then passes to decision 316 to determine whether the SMB contained a file close instruction. If so, control returns to item 302. If not, control is passed to decision 304 described above.

BNSDOCID: <WO__9103024A1_I_>

Figure 4 is a flow diagram which describes the operation of the workstations of the present invention when requesting a file to be opened in the OPlocked exclusive mode. The routine 400 is invoked whenever a client requests operation in a sharing mode as indicated by item 402. Control then passes to item 404 wherein the workstation redirector converts this request to a request to open a file in the exclusive OPlocked mode, generates the appropriate SMB and sends the SMB to the file server via the LAN communication lines. Control then passes to item 406 wherein the workstation waits for a reply. Once a reply is received, control passes to item 407 to decode the SMB message from the server. Decision 408 then determines whether the OPlock request was granted. If not, control passes to item 410 wherein the file is open in a normal non-buffered mode. Control then passes to loop 412 which comprises item 414 and decision 416. Item 414 is invoked to perform system I/O until the file is closed. Decision 416 determines whether the file is closed. If decision 416 determines that the file is not closed, control loops back to item 414. If decision 416 determines the file is closed, control passes to item 418 which indicates the file is closed.

If decision 408 determines OPlock was granted, control passes to item 420 which indicates the file is open in the OPlocked exclusive mode. Control then passes to item 422 which allows buffered operations such as read ahead, write behind and lock caching. Loop 424 is then invoked. Loop 424 comprises item 424, which performs buffered I/O on the open file until it is closed. Decision 428 determines whether the file is closed. If decision 428 determines that the file is not closed, control loops back to item 422. If decision 428 determines the file is closed, control passes to item 418, which indicates the file is closed. Note: loop

424 can be modified if an asynchronous break_oplock request (detailed in Figure 5) is received.

Figure 5 is a flow diagram which describes the operation of the workstations of the present invention when the OPlocked exclusive mode operation must be terminated. The process 500 is invoked when a break_oplock SMB is received from the file server as indicated by item 502. Decision 504 then determines whether the referenced file is open. If not, the break_oplock SMB is ignored in item 506. If the referenced file is open, control passes to item 508 which transmits any write behind data and cached locks to the server and which invalidates any read ahead data. Control then passes to item 510 which transmits an oplock_broken SMB to the file server to indicate OPlock is broken and the workstation reverts to the sharing mode in item 512 by changing the file status to operate in a non-buffered I/O mode.

Based on the description above, it will be readily apparent to those of ordinary skill in the art that the present invention may be implemented with any of a number of well known LAN protocols. However, the present invention is particularly adapted for use with the Server Message Block (SMB) file sharing protocol. The SMB file sharing protocol is described in detail in a document "Microsoft Networks/OpenNET File Sharing Protocol," Intel part Number 138446, Microsoft Corp., 1985, attached hereto as Appendix I, and a document "SMB File Sharing Protocol Extensions," Microsoft Corp., 1988, attached hereto as Appendix II, both of which are incorporated herein by reference. References to the OPlocking technique of the present invention may be found in Appendix II, pp. 3, 4, 46-48 and 51-58. It should be noted that the references to the present invention is the document attached hereto as Appendix II were printed less than one year prior to the filing date of the present application.

BNSDOCID: <WO__9103024A1 I >

In summary, a method and means for improving the performance of a distributed computer system has been described. In accordance with the principles of the present invention, a distributed computer system comprises a plurality of workstations and a file server. The workstations of the present invention are adapted to operate in a sharing mode, an exclusive mode and an OPlocked exclusive mode. In the sharing mode and exclusive mode, the workstations of the present invention operate in a conventional manner. However, in the Oplocked exclusive mode, requests for opening file server files in the sharing mode are converted automatically into requests for opening files in the Oplocked exclusive mode. If the requested file has not already been opened by another workstation, the file open request is granted for Oplocked exclusive mode operation wherein the file or portions of the file are transferred from the file server to the requesting workstation where it is cached and processed locally. If another workstation generates a request to open an Oplocked file, the workstation holding the file open in the Oplocked exclusive mode transfers its cached data back to the file server and reverts to the conventional sharing mode. Accordingly, other uses and modifications of the present invention will be readily apparent to persons of ordinary skill in the art without departing from the spirit and scope of the present invention. All of such uses and modifications are intended to fall within the scope of the appended claims.

BNSDOCID: <WO__ 9103024A1_I_>

## Claims

1. An improved local area network comprising:

file server means for providing file services to a plurality of clients wherein said file services include storing files and providing access to said files for reading and modifying said files;

a plurality of remote workstations wherein said remote workstation includes means for locally processing and caching data and further wherein said workstations are adapted for operation in sharing, exclusive and Oplocked exclusive modes;

means for automatically opening files in said Oplocked exclusive mode when one of said remote workstation request access to file server files and no other remote terminal currently has the requested file open; and

means for reverting from said Oplocked exclusive mode to said sharing mode if another workstation requests access to a file opened in said Oplocked exclusive mode.

2. A method of improving system performance in a local area network including a file server and a plurality of remote workstations, comprising the steps of:

(a) issuing a request from one of said remote workstations to open a file resident on said file server in a sharing mode;

(b) automatically converting said sharing mode request into a request to open said file in an Oplocked exclusive mode;

(c) determining whether the requested file has been opened by another workstation;

(d) opening said file in an Oplocked exclusive mode if said requested file is not currently open;

(e) opening said file in a sharing mode if said said requested file is currently open; and

BNSDOCID: <WO___9103024A1_I_>

(f) reverting to said sharing mode if said file is opened in said Oplocked exclusive mode and the file is requested by another of said workstations.

3. A method of improving system performance in a local area network including a file server and a plurality of remote workstations, comprising the steps of:

(a) means for issuing a request from one of said remote workstations to open a file resident on said file server in a sharing mode;

(b) means for automatically converting said sharing mode request into a request to open said file in an Oplocked exclusive mode;

(c) means for determining whether the requested file has been opened by another workstation;

(d) means for opening said file in an Oplocked exclusive mode if said requested file is not currently open;

(e) means for opening said file in a sharing mode if said said requested file is currently open; and

(f) means for reverting to said sharing mode if said file is opened in said Oplocked exclusive mode and the file is requested by another of said workstations.

BNSDOCID: <WO__9103024A1_I_>

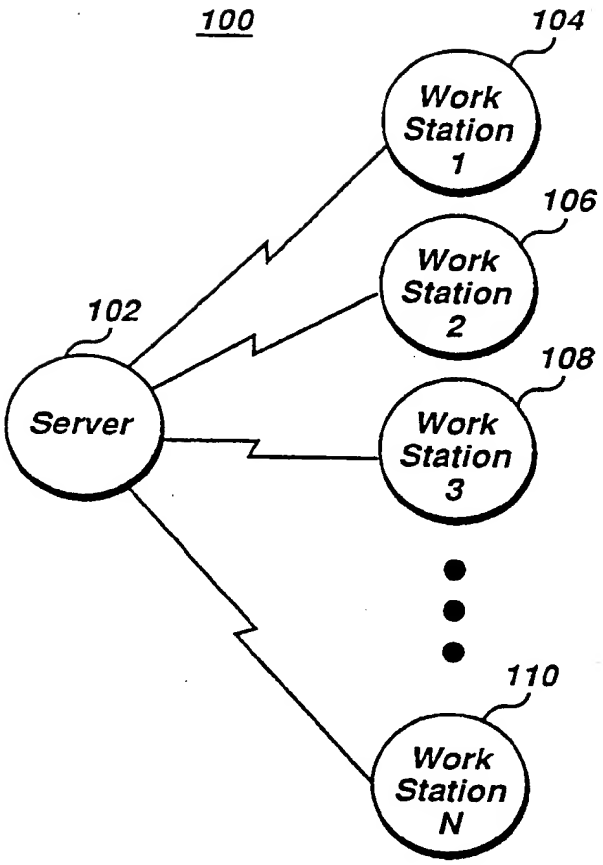



Fig. 1



BNSDOCID: <WO___9103024A1_I_>

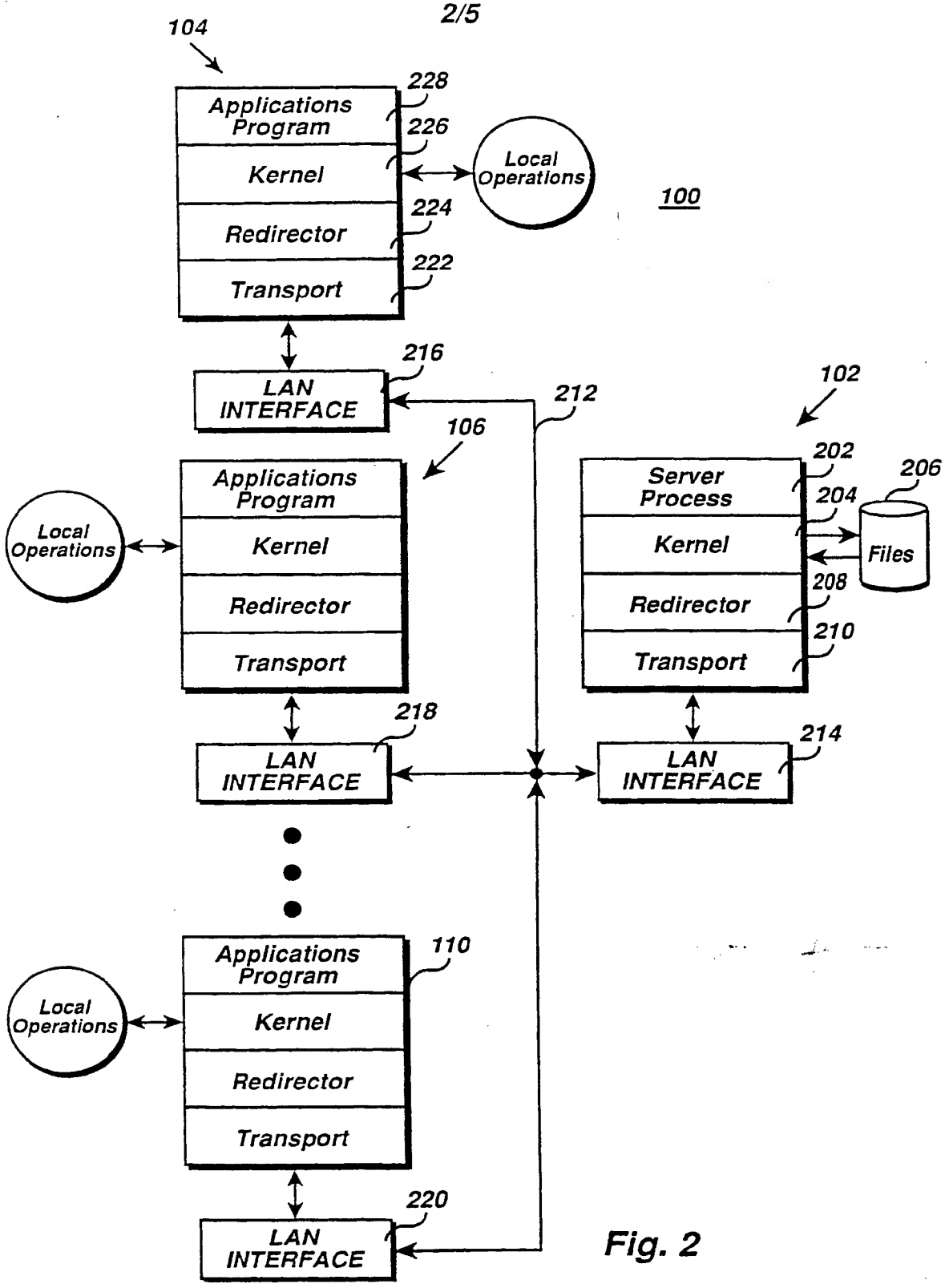



Fig. 2

SUBSTITUTE SHEET

BNSDOCID: <WO__9103024A1_I_>

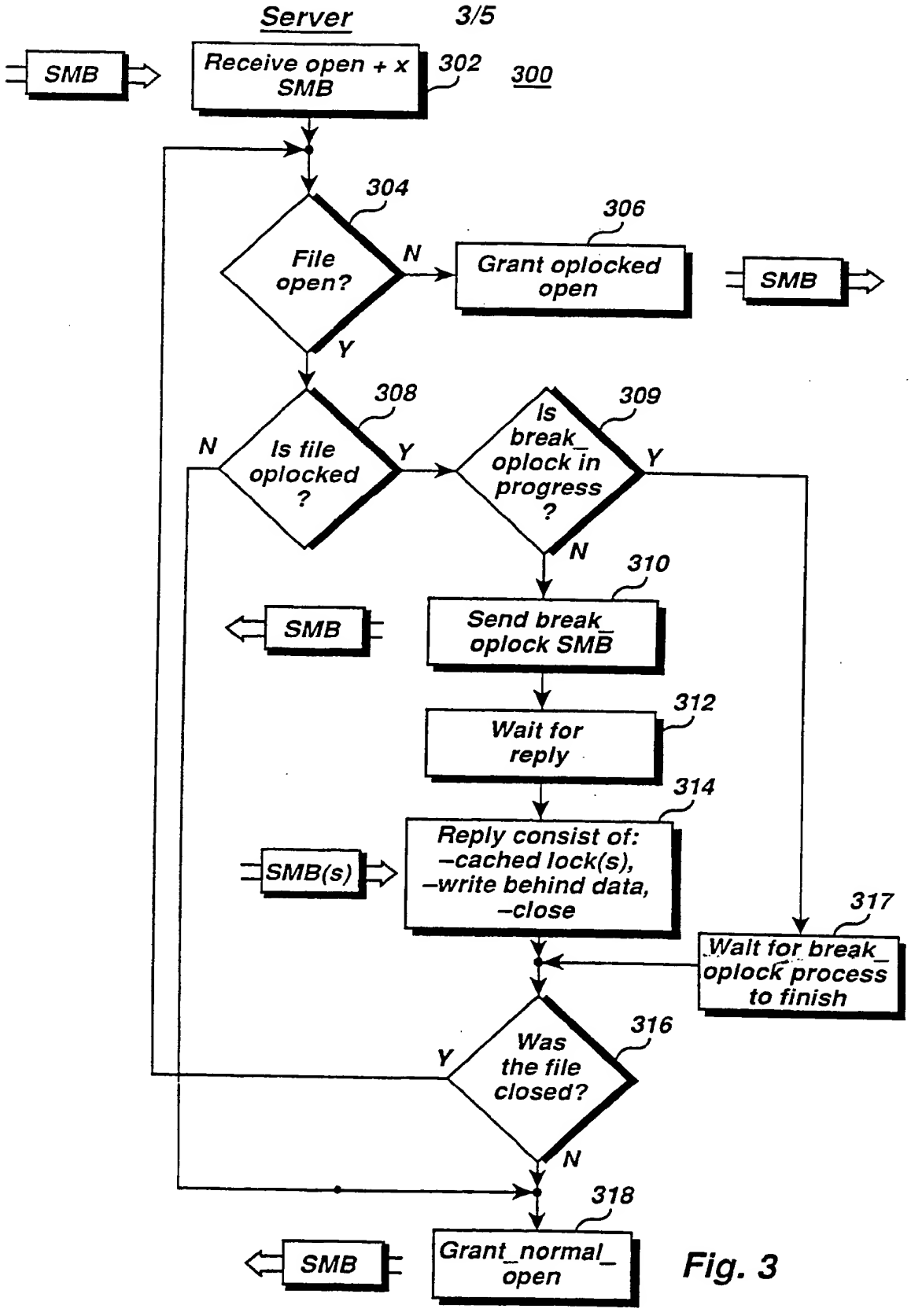



Fig. 3

SUBSTITUTE SHEET

BNSDOCID: <WO__9103024A1_I_>

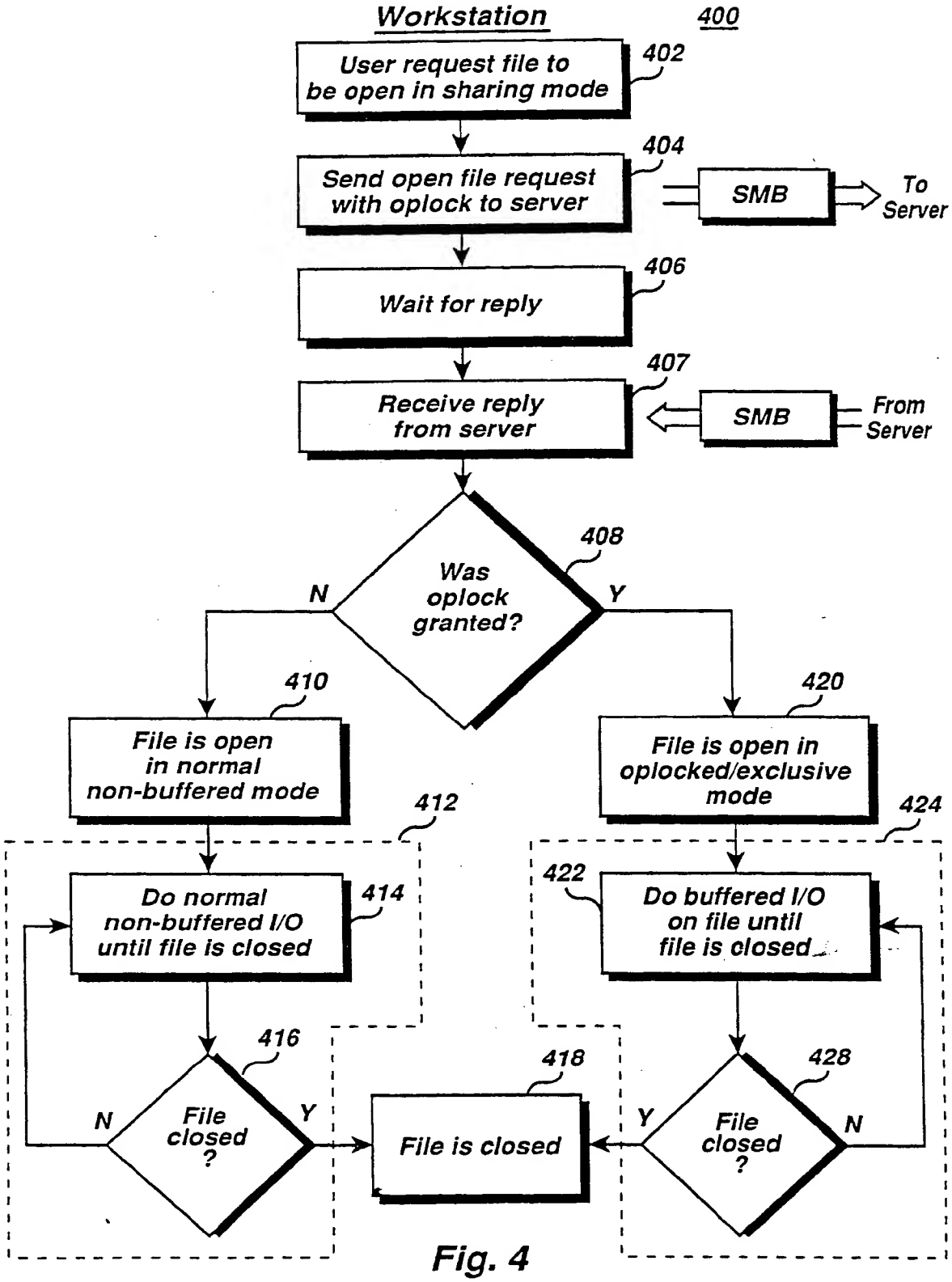



Fig. 4

SUBSTITUTE SHEET

BNSDOCID: <WO___9103024A1_I_>

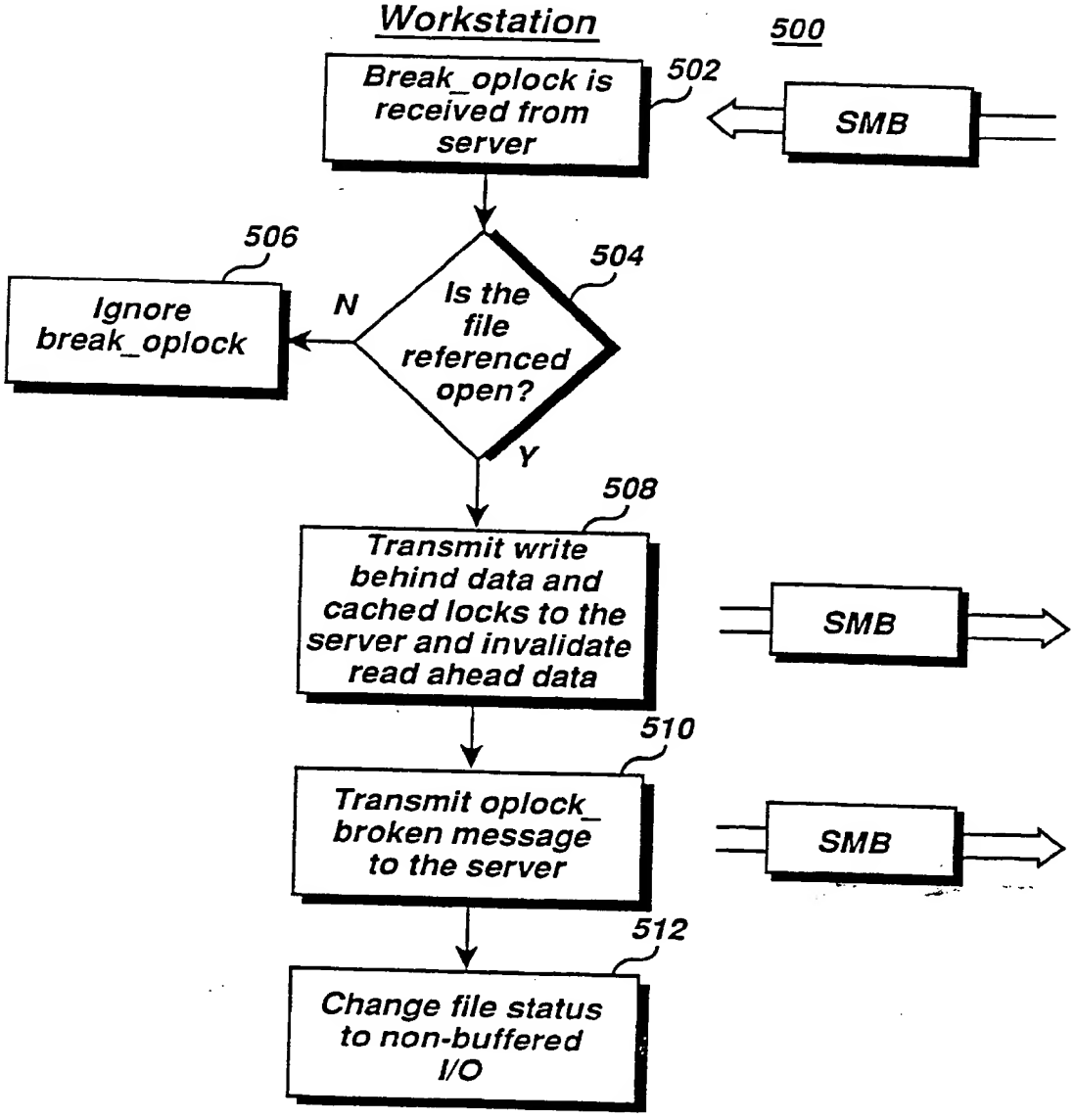



Fig. 5



NSDOCID: <WO___9103024A1_I_>

# INTERNATIONAL SEARCH REPORT

International Application No PCT/US 90/04570

## I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) [6]

According to International Patent Classification (IPC) or to both National Classification and IPC

IPC[5]: G 06 F 15/403, G 06 F 15/40

## II. FIELDS SEARCHED

Minimum Documentation Searched [7]

| Classification System | Classification Symbols |
|---|---|
| IPC[5] | G 06 F |

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched [8]

## III. DOCUMENTS CONSIDERED TO BE RELEVANT [9]

| Category [10] | Citation of Document, [11] with indication, where appropriate, of the relevant passages [12] | Relevant to Claim No. [13] |
|---|---|---|
| A | EP, A, 0278315 (IBM) 17 August 1988 see abstract, page 5, line 55 - page 6, line 40; page 16, line 40 - page 18, line 23; figures 6,7 -- | 1-3 |
| A | Communications of the ACM, volume 29, no. 3, March 1986, (New York, NY, US), J.H. Morris et al.: "Andrew: A distributed personal computing environment", pages 184-201 see page 189, right-hand column, line 13 - page 190, right-hand column, line 54, figure 1 -- | 1-3 |
| A | Proceedings of the Tenth ACM Symposium on Operating Systems Principles, 1-4 December 1985, Orcas Island, Washington, US, S.J. Mullender et al.: "A distributed file service based on optimistic concurrency control", pages 51-62 see sections 4, 4.3, 4.5 | 1-3 |

* Special categories of cited documents: [10]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

## IV. CERTIFICATION

| Date of the Actual Completion of the International Search | Date of Mailing of this International Search Report |
|---|---|
| 9th January 1991 | 28 JAN 1991 |
| **International Searching Authority** | **Signature of Authorized Officer** |
| EUROPEAN PATENT OFFICE | Mme N. KUIPER |

Form PCT/ISA/210 (second sheet) (January 1985)

BNSDOCID: <WO__9103024A1_I_>

# ANNEX TO THE INTERNATIONAL SEARCH REPORT ON INTERNATIONAL PATENT APPLICATION NO. US 9004570 SA 39908

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report. The members are as contained in the European Patent Office EDP file on 18/01/91 The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| EP-A- 0278315 | 17-08-88 | US-A- 4887204 | 12-12-89 |
| | | JP-A- 63200244 | 18-08-88 |

EPO FORM P0479

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

BNSDOCID: <WO___9103024A1_I_>

THIS PAGE BLANK (USPTO)